VERIFICATION OF GRAPHEMES USING NEURAL NETWORKS IN AN HMM-BASED ON-LINE KOREAN HANDWRITING RECOGNITION SYSTEM

So, S.J.; Kim, J.; Kim, J.H.

Published in:
EPRINTS-BOOK-TITLE

VERIFICATION OF GRAPHEMES USING NEURAL NETWORKS IN AN HMM-BASED ON-LINE KOREAN HANDWRITING RECOGNITION SYSTEM

SUNG J. CHO, JAHWAN KIM, AND JIN H. KIM

Department of Electrical Engineering & Computer Science. KAIST
373-1, Kusong-dong, Yusong-ku,Taejon, 305-701
Korea
E-mail: {sjcho,jahwan, jkim}@ai.kaist.ac.kr

This paper presents a neural network based verification method in an HMM-based on-line Korean handwriting recognition system. It penalizes unreasonable grapheme hypotheses and complements global and structural information to the HMM-based recognition system, which is intrinsically based on local information. In the proposed system, each grapheme has one neural network verifier as well as one HMM recognizer. The verifier takes as an input the grapheme hypothesis generated by the HMM and outputs *a posteriori* probability as its validity. This probability is then incorporated into the search process by Viterbi algorithm during recognition. The global and structural information to the verifier is obtained from the relationship between primitive strokes in each grapheme by analyzing their correspondence with the HMM states. The experimental result shows that the recognition error of the baseline HMM network can be reduced by 39.2% with the proposed verification scheme.

1 Introduction

In this paper, verification means to validate hypotheses generated by HMM recognizers during the recognition process. There are at least two purposes for its use. The first is to penalize the unreasonable grapheme/alphabet hypotheses as early as possible, generated by the HMM recognizers [5]. The second purpose is to complement the HMM recognizers which miss global and structural information due to the first order Markov assumption [4,9].

A number of studies have been proposed that use postprocessing and verification steps in HMM based recognition systems. The duration distributions of HMM states in the most probable path were used for postprocessing [3,4,7]. The portions of a handwriting input corresponding to the HMM states were also used for postprocessing by measuring various statistics between them [8]. Another system used DP matching for complementing global information to the HMM recognizers [6]. Statistical grammar rules were also used for verification [5]. These mechanisms reduced recognition errors by about 15~48% [5,8].

Despite their success in reducing errors, the above approaches have two problems when applied to the previous HMM based recognition system for on-line

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 219-228

Korean handwriting [1]. First, they lack systematic integration scheme of the verification result and the HMM probability. The verification needs various sources of information. However, if they are expressed as separate probability distributions, the overall verification probability drops down rapidly as their numbers increase because of probability multiplication [5,8]. Furthermore, it is not also easy to normalize them according to their importance and reliability. This is also true when DP matching is used for verification [6]. Second, they have limitation in extracting global and structural features reliably. These features are especially important in the recognition of Korean characters [10] whose graphemes are structurally constructed from primitive strokes such as line segments and circles. However, the previous approaches such as utilization of duration statistics in the HMM states [8], the attributed grammars [5] or DP matching methods [6] have limitation to reflect them.

In order to remedy the integration problem, we propose a neural network verifier in this paper. Each grapheme has one HMM recognizer and one neural network verifier. The HMM recognizer generates a grapheme hypothesis, *i.e*, the position of the grapheme in the given feature sequence and its likelihood. The verifier then approximates *a posteriori* probability of its validity. This probability is then incorporated into the HMM network during the Viterbi search. All the measurements for the verification are utilized in our study as feature inputs to the neural network, which were usually represented as the separate probability distributions in the previous studies. Consequently, regardless of how many measurements are used, the verification result is represented by only one *a posteriori* probability. Their normalizations are also done by the neural network weights.

In order to extract structural and global features in a grapheme, we use the relationship between its primitive strokes. The position and the length of each primitive stroke are used for the features in the paper. The primitive strokes are extracted by analyzing their correspondence with the HMM states after Viterbi search. Our past experience on HMM system shows that each state in a HMM recognizer models a primitive stroke reliably. The notion of macro states [8] is also a similar idea.

Our experimental results on the Korean character data show that our method effectively reduced the recognition error compared to the baseline HMM system. From this result, we may conclude that rich information from the global and structural features is successfully incorporated in our hybrid framework of the HMM recognizer and the neural network verifier.

The present paper is organized as follows. In section 2, we discuss the necessity and the integration scheme of a neural network verifier to the HMM recognizer. In section 3, we explain the structural and global features for the verifier and how they are obtained from the correspondence between the primitive strokes and the HMM states. In section 4, the experimental results are described. Conclusion then follows in section 5.

2 Neural Network Verification System

2.1 Advantages of a neural network as a verifier

The purpose of a verifier is not to recognize a grapheme but rather to determine whether a grapheme hypothesis generated by a baseline HMM recognizer is valid or not. This means that an incorrect hypothesis should be penalized while a correct one passed intact. It should also complement the weakness of the HMM recognizer. Therefore, a verifier should satisfy the following properties.

1. The verification result should be systematically integrable with HMM probability. The verifier uses various sources of information. Therefore, adding information should not affect the overall verification scheme. Probability is a preferable representation of a verification result because an HMM network uses Viterbi search which finds a maximally probable path.
2. The verifier should complement the HMM recognizer by using global and structural features. The HMM usually uses local features only, thus missing global trend information. It is because all observation vectors are assumed to be independent of one another by the first order Markov assumption.
3. The verifier should be able to discriminate similar shapes. Many Korean graphemes, especially vowels, can only be discriminated based on a small but critical portion of a handwriting input. HMM has difficulty in this regard because of the maximum likelihood estimation (MLE) training method.

So far, no method has been proposed that satisfies all these properties. When DP matching is used [6], the probability of HMM is mixed with non probability values, *i.e*, distance. This makes it difficult to normalize the distance and probability to the same scale. When measurement statistics are used such as distribution of duration, distance between portions of handwriting input and structural features, the overall verification probability decreases as the number of them increases [5,8], because of multiplication of several probability terms. Furthermore, it is not easy to normalize their scales according to their importance and reliability. Also, none of the proposed methods have discrimination capability except pairwise discrimination method [5], in which critical features for confusing pairs are manually pre-specified.

A neural network verifier satisfies all these criteria. The verification result can be smoothly integrated into the HMM probability in the probabilistic framework because the neural network is known to model *a posteriori* probability if properly configured [2]. The global and structural features are represented as feature inputs of the neural network. Therefore, overall verification probability doesn't decrease as more measurements are employed. The error backpropagation training algorithm makes it possible to discriminate similar graphemes by giving larger weights to the critical features. These reasons drive us to choose a neural network as the verifier in the proposed study.

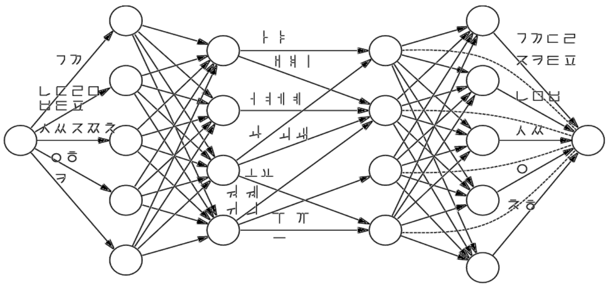

Figure 1. The baseline HMM network(‘BongNet’) [1]. Each arc represents a grapheme or a ligature HMM recognizer.

2.2 Baseline HMM network for Korean handwriting recognition

A typical Korean character (Hangul) is structurally constructed from one first consonant, one vowel and one optional last consonant. There are 19 first consonants, 21 vowels and 27 last consonants. Therefore, the recognition of a Hangul character resembles that of an English word because an English word is also composed of alphabets, only more difficult because of the way those graphemes are combined two-dimensionally.

The baseline system was an HMM network previously reported in [1]. The graphemes and ligatures between them were modeled as separate discrete left-to-right HMMs. The number of states in each HMM was different, depending upon the complexity of the grapheme. The feature was a simple chaincode sequence with sixteen directions and a pen up/down status. These grapheme and ligature models were connected according to the composition rule of a Hangul character as shown in Figure 1. A path from the leftmost node to the rightmost node determines a unique character.

2.3 Integration of the neural network verifiers with the HMMs

Each grapheme has one HMM recognizer and one neural network verifier as shown in Figure 2(b). The HMM generates grapheme hypotheses during recognition. It gives the portion of the grapheme in the handwriting input sequence and the duration in each HMM state. From this, the global and structural features for the verifier are extracted. Further details are described in the next section.

We use as a verifier a multi-layer perceptron(MLP) with one input, one hidden and one output layers. It uses a logistic function for nonlinearity and has one output node. The value of the output node becomes one when the hypothesis under test is correct and zero otherwise. The desired output is one if the input belongs to the grapheme and zero otherwise. Under this structure and backpropagation training algorithm, it is known that the activation value of the output node approximates

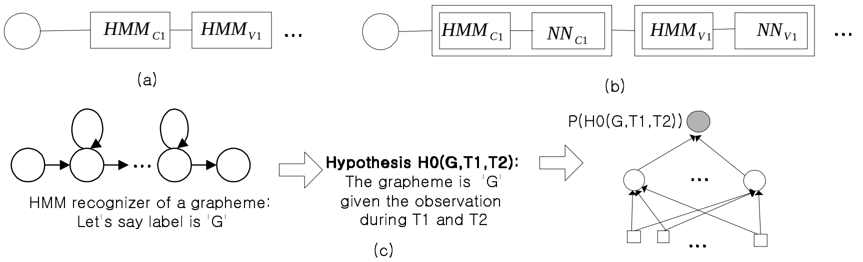

Figure 2. (a) A path in the baseline HMM network (b) The neural network verifier follows the HMM in each grapheme. (c) The structure of the HMM and the verifier. the HMM gives hypothesis to the verifier. Then verifier approximates *a posteriori* probability of its validity.

a *posteriori* probability that the hypothesis is valid [2].

This *a posteriori* probability is integrated with the probability of HMM network in probabilistic framework. Figure 3 shows how this can be done. The final result shows that the activation value of the output node of the neural network is multiplied with the HMM probability.

λ_g : The HMM recognizer of the grapheme g

N_g : The neural network verifier of the grapheme g

O_g : The portion of a handwriting input aligned to the grapheme g

$\Gamma(\lambda_g, O_g)$: The duration and partitioned O_g by each HMM state after O_g is aligned to λ_g by Viterbi search

$N_g(x)$: The activation value of the output node when x is given

$$P(\lambda_g, N_g \mid O_g) = P(\lambda_g \mid O_g) P(N_g \mid \lambda_g, O_g)$$

$$= P(\lambda_g \mid O_g) P(N_g \mid \Gamma(\lambda_g, O_g)) = \frac{P(\lambda_g) P(O_g \mid \lambda_g)}{P(O_g)} P(N_g \mid \Gamma_g(\lambda_g, O_g))$$

$$= \frac{P(\lambda_g)}{P(O_g)} P(O_g \mid \lambda_g) N_g(\Gamma_g(\lambda_g, O_g)) \approx P(O_g \mid \lambda_g) N_g(\Gamma_g(\lambda_g, O_g))$$

($\because \frac{P(\lambda_g)}{P(O_g)}$ is assumed to be same for all the graphemes)

Figure 3. Integration of *a posteriori* probability of the NN verifier and the HMM probability

3 Features for Neural Network Verifier

3.1 Primitive strokes

The shape of each Korean grapheme is structurally constructed from its primitive strokes. There are two sorts of primitive strokes as shown in Figure 4. One

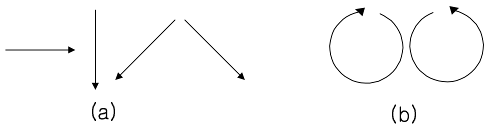

Figure 4. Primitive strokes in Korean graphemes. Almost all the graphemes have straight lines only.

(a) Straight lines (b) Circles

is a straight line and the other is a circle. Almost all the graphemes have straight lines only. Only four of them contain circles. This structural composition of the primitive strokes gives a foundation of structural analysis of graphemes [10].

However, it is difficult to extract them reliably, since they are modified and connected smoothly in a handwritten grapheme similarly to the coarticulation phenomena in the speech [3]. Therefore, simple segmentation methods such as finding extreme points do not work well.

To extract them reliably, we use the correspondence between them and the HMM states. In this paper, a primitive stroke is defined as the portion in a handwriting input aligned to the corresponding HMM states after Viterbi search. It is also similar to the idea in [8] in which the HMM states were used for extracting structural information. Figure 5 shows examples of this correspondence observed in the training data. It shows that the primitive strokes from the corresponding HMM states coincide with our conceptual ones.

From such a mapping table, they can be easily extracted. For example, in the grapheme data 'ㄱ', the horizontal stroke can be obtained by extracting the portion in the handwriting input aligned to the HMM state 1.

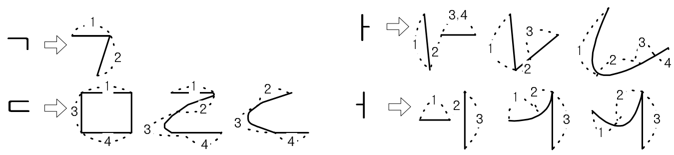

Figure 5. Examples of mapping between the primitive strokes and the HMM states. The grapheme labels are shown on the left of the arrows and the typical data aligned to the HMM states are shown on the right. The small digits near the pen strokes represent the HMM state number matching them. These are observed in the training data.

3.2 Details of the features

In the present system, three different features are used based on these primitive strokes. They are the position of the primitive strokes (structural feature), the duration in each HMM state and the accumulated direction change (global feature).

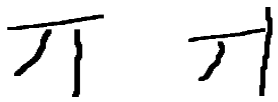

Figure 6. The above two vowels have the same primitive strokes and writing order, but they differ in the position of the rightmost vertical lines. The left vowel is ‘ㅠ’ and the other is ‘ㅟ’.

The position of the primitive strokes is important in Korean graphemes, especially in the vowels. This is illustrated in Figure 6. Both vowels have the similar primitive strokes and the writing order. The only difference between them is the position of the rightmost vertical lines. For this reason, we use their position as the structural feature. It is represented by the starting and ending point of each primitive stroke normalized by a bounding box. Figure 7(a) shows an example of extracting the feature.

One of the global features is the duration information in each HMM state, which is known to be useful complementary information [3,4,7]. It is because that the conventional HMM models the duration in each state with unrealistic exponential distribution but the postprocessing step gives more realistic duration distribution. The duration is defined as the number of chaincodes consumed in each HMM state in this paper. It can also be interpreted as the length of the corresponding primitive stroke. It is shown in figure 7(b).

Another global feature is the accumulated direction change. It is the sum of direction changes in a handwriting input. It reflects the global orientation of the pen movement and measures the complexity of a grapheme because it is large in the complex grapheme. Clockwise and counterclockwise direction changes are summed separately, as shown in the figure 7(c).

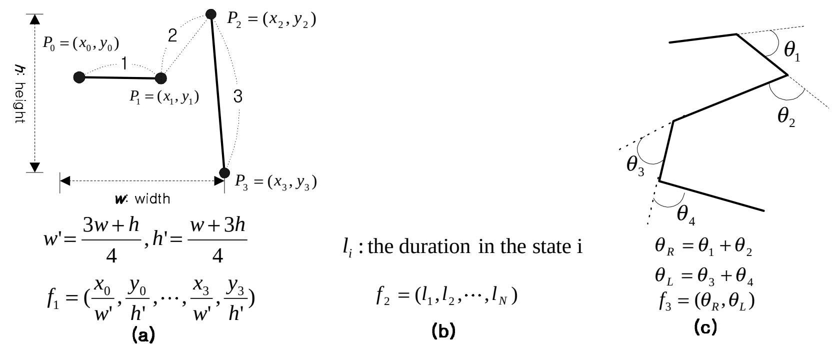

Figure 7. Three features used in the neural network verifier:

(a) Position of primitive strokes, (b) Duration in each state, and (c) Accumulated direction change

4 Experimental Results

4.1 Data set

The data[1] for the experiment were collected from high school and college students. There was no restriction or guidance in the writing styles. As a result, cursive writing style as well as run-on style were found in the data.

For training, graphemes were extracted manually from characters as there was no indication of any explicit grapheme boundary in the data. The training data had 49,049 characters written by 48 writers.

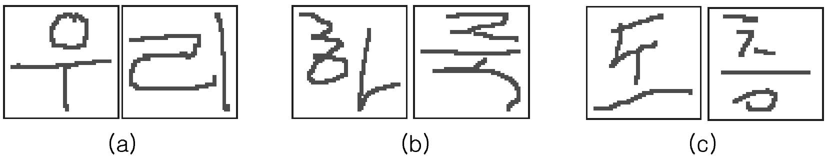

Figure 8. Examples of the test data. (a) Nation (b) High School (c) JungAng

The test data consists of three separate sets, labeled 'Nation', 'High School' and 'JungAng'. The different writing styles were observed in the data sets as shown in Figure 8. By subjective judgment, their writing styles were classified and shown in Table 1.

Table 1. Properties of the test data. Writers of the test data were different from those of the training data.

	Nation	High School	JungAng
Writers	9	9	39
Characters	3,127	15,250	16,427
Writing style	well written	highly variable, cursive	normally written

4.2 Neural network training

The neural network was trained by the backpropagation algorithm. Each grapheme had two kinds of data set. One was the positive data set composed of that grapheme. The other was the negative data set composed of all but the positive data. When training the MLP, the desired output for the positive data was 1, whereas that for the negative data was 0.

The ligatures, *i.e*, the connection strokes between graphemes were not verified in this system. We believed that their shapes were too much varied to be verified effectively, especially in the cursive writing style.

To assess performance of the verifiers, errors were categorized into two types.

[1] Half of the data is available from our web page(http://ai.kaist.ac.kr/)

1. Type I error: The handwriting input belongs to the grapheme, but the verifier answers negatively. This error is somewhat critical because the correct hypothesis is penalized.
2. Type II error: The handwriting input doesn't belong to the grapheme, but the verifier answers positively. This error is not so critical compared to the type I error because the correct hypothesis is intact and the baseline system may still be able to discriminate the hypotheses.

Table 2 showed the recognition results in terms of above errors for the training data. The verifiers were well trained with less than 0.8 % error rate for all the data. However, type I error rates were always more than type II error rates. We believe that this was due to the lack of positive data.

Table 2. Type I and II errors in the training data set. This result shows that the verifiers were well trained. (When *a posteriori* probability of the positive data was less than 0.5, then we regarded it as type I error. Also when that of the negative data was more than 0.5, then we regarded it as type II error.)

	First Consonant	Vowel	Last Consonant
Type I errors	0.78%	0.65%	0.70%
Type II errors	0.30%	0.21%	0.21%

4.3 Recognition test

To examine the advantage of our neural network verifier, we performed the grapheme recognition test and the complete character recognition test.

In the grapheme recognition test, the neural network verifier reduced errors by about 59.9% for the training data as shown in the Table 3. This result suggested that it was well trained and had reliable discrimination power.

Table 3. The grapheme recognition rate of the baseline HMM network and the neural network verifier for the training data

	First Consonant	Vowel	Last Consonant	Average
Baseline	94.83%	89.90%	91.92%	92.22%
Baseline + NN verifier	97.65%	96.55%	96.44%	96.88%

Because the graphemes were more correctly classified, the recognition rate of a complete character also became higher as shown in Table 4. The errors were reduced across all the data sets. The overall error reduction was about 39.2%.

Table 4. The complete character recognition rate of the baseline HMM network and the neural network verifier for the test data

	Nation	High School	JungAng	Average	Error reduction
Baseline	91.07%	89.77%	88.36%	89.73%	-
Baseline + NN verifier	95.11%	93.81%	92.37%	93.76%	39.2%

5 Conclusion

A neural network verifier was developed to complement the baseline HMM network. The grapheme hypothesis generated by the HMM recognizer was verified with the structural and global features based on the correspondence between the primitive strokes and the HMM states. *A posteriori* probability of its validity from the verifier was then merged with the probability of the HMM. The neural network made it easy to systematically utilize various source of information for the verification. The experimental results showed that this verification scheme reduced the error of the baseline HMM recognizer by 39.2% in the Korean character recognition.

References

1. Bong-Kee Sin and Jin H.Kim, "Ligature Modeling for Online Cursive Script Recognition," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, June 1997, pp.623-633
2. M.D.Richard and R.P.Lippmann, "Neural Network Classifiers Estimates Bayesian *a posteriori* Probabilities," *Neural Computation*, Vol.3, 1991, pp.461-483
3. L.R.Rabiner ,and B.-H.Juang, *Fundamentals of Speech Recognition*, Prentice Hall ,1993
4. L.R.Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of IEEE, Vol.77, No.2, pp.257-285, Feb. 1989
5. Jae-Ook Kwon, Bong-Kee Sin, and Jin H. Kim, "Recognition of On-line Cursive Korean Characters Combining Statistical and Structural Models," *Pattern Recognition*, 30, 8(1997), pp 1255-1263
6. J.Hu, S.G.Lim, and M.K.Brown, "HMM Based Writer Independent On-line Handwritten Character and Word Recognition," *Proc. of the Sixth IWFHR*, Taejon, Korea,1998, pp 143-155
7. A.Senior, J.Subrahmonia, and K.Nathan, "Duration Modeling Results For an On-line Handwriting Recognizer," *Proc. of Int. Conference on Acoustics, Speech and Signal Processing*, 1996, Vol.6, pp3483-3487
8. Jinhai Cai and Zhi-Qiang Liu ,"Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 3, March 1999
9. J.G.A. Dolfing and R.Haeb-Umbach, "Signal Representations For Hidden Markov Model Based On-line Handwriting Recognition," *Proc. of Int. Conference on Acoustics, Speech and Signal Processing*, 1997, pp3385-3388
10. H.Y.Kim and Jin.H.Kim, "Hierarchical Random Graph Representation of Handwritten Characters and its Application to Hangul Recognition," *Pattern Recognition*, (in press)